Concepts from data

Brandon Rohrer*, **J. Dan Morrow, Fred Rothganger, and Patrick G. Xavier**
Sandia National Laboratories
PO Box 5800, MS 1010
Albuquerque, New Mexico 87185

Abstract

Creating new concepts from data is a hard problem in the development of cognitive architectures, but one that must be solved for the BICA community to declare success. Two concept generation algorithms are presented here that are appropriate to different levels of concept abstraction: state-space partitioning with decision trees and context-based similarity.

Introduction

The challenge of developing novel concepts from raw sensory data is central to the effort to make intelligent machines. The majority of cognitive architectures begin with a predefined set of concepts (also referred to as categories, percepts, classes, or domains). The set of concepts is typically selected by the programmer and fixed. However, the ability to create new concepts from a set of observations is a key cognitive ability. It allows adaptation to new environments and tasks, re-interpretation of past experiences, and increasingly abstract processing. It is arguable that the goal of machine cognition is unobtainable without this capability.

Previous work directed at creating concepts based on observed data (also termed conceptual clustering) includes the CLUSTER/2 algorithm, COBWEB, and UNIMEM. CLUSTER/2 uses a method called conjunctive conceptual clustering to produce a partition of the input state space. The partitioning on a data set is optimized on a number of criteria, with an emphasis on simplicity. (Michalski and Stepp 1983) COBWEB is somewhat similar, optimizing the partitioning on the extent to which the partition allows prediction of individual attribute values within each concept. (Fisher 1987) UNIMEM classifies specific observations into a hierarchy of generalization, based on their individual attribute values. (Lebowitz 1987) COBWEB and UNIMEM are notable for being incremental, that is, they incorporate new data as they are received.

As the term is used here a "concept" is a set of numerical or symbolic data that are closely related. More specifically, the data within a single concept are, to a certain extent, interchangeable. A reference to a concept may equally well be referring to any member of the concept set. In this way, concepts allow a granularity of representation that can greatly reduce the size of the state space required to represent a data set.

For instance, the mass of an apple may be measured with arbitrary accuracy and thus may be represented by a real number. This variable may take on infinitely many values, even if given an upper and lower bound for the mass. However, the apples may be classified into groups based on the concepts *big* and *small*, reducing the size of the state space to just two possible values.

A similar process can be followed with symbolic data. The symbols (more specifically in this example, strings) *apple*, *orange*, *banana*, and *grape* could be referenced interchangeably in some cases. The concept of *fruit* to which these all belong allows the whole set of them to be referred to generically.

It should be noted that interchangeability is just one way that data can be related. Nearness (co-occurrence, adjacency, or temporal proximity) is another. Co-occurrence is used as the basis for relatedness in the majority of natural language processing approaches. A prominent example of this is the bag-of-words mechanism that underlies Latent Semantic Analysis (LSA) (Landauer and Dumais 1997), an algorithm that identifies concept-like sets of terms. A related approach is the fixed-length window used in the Hyperspace Analog to Language (HAL) model of sematic distance. (Lund, Burgess, and Atchley 1995) CBS, however, is based on substitutability, a similarity criterion espoused by some of the earliest practitioners of computational linguistics (Rubenstein and Goodenough 1965; Miller and Charles 1991) and used in some current efforts as well (McCallum 1996; Wang, McCallum, and Wei 2007). An approach extremely similar to CBS was published in 2006 (Carbonell et al. 2006), just one year after the first published description of CBS (Hulet, Rohrer, and Warnick 2005).

In this paper we describe two approaches to the problem of deriving new concepts from data. Each has various strengths and likely areas of applicability. The first, state space partitioning with decision trees, is appropriate to creating concepts out of low level data that may be numerical, symbolic, or a combination of the two. The second, context-

*Corresponding author. Email: brrohre@sandia.gov, Internet: http://sites.google.com/site/brohrer/

based similarity (CBS), is suited to building concepts out of symbolic data, specifically a serial stream of symbols.

Method I: State-space partitioning with decision trees

Decision trees (Bentley 1975; Breiman et al. 1984; Quinlan 1987) can be used to partition the state space, mapping regions of the state space onto a hierarchy of concepts. It creates concepts by subdividing the input state space of the cognitive system. Each region created by dividing the state space represents a separate concept. Those areas of the state space in which the most observations occur are subdivided most finely, concentrating representational resources where they are most needed. The resulting hierarchy of divisions is represented as a decision tree. This approach is best suited to low-level data in parallel streams, although it may be applied to any form of data, and it operates on-line and in real-time.

In order to facilitate the creation of a decision tree, the input is represented as a vector. Each element of the vector may be valued appropriate to the data type it represents. Continuous data is represented with real numbers. Either integers of reals can be used to represent discrete data. Integers may also be used to represent ordinal data or membership in one of several categories. Symbols and attribute data may be represented with binary values. The exact form of the representation is not critical, as long as every distinct set of data values can be mapped to a unique numerical vector. The set of all possible values of the vector defines the extent of the state space. The number of elements in the vector define the number of state space dimensions.

Concept creation through subdivision

There are (at least) two ways to autonomously create partitions within the state space: through agglomerative clustering or through subdividing. In clustering, all the individual points in the state space are initially treated as separate concepts, then placed into larger groups representing more general concepts over time. In subdividing, the state space is treated as one super-concept, which is repeatedly divided into finer and finer-grained concepts. This second approach is used here, with a variant of a decision tree representing the hierarchy of divisions.

The algorithm for bisecting the state space has only to decide when to divide it and along which dimension. The method used is as follows:

1. Accept a new state observation.
2. Traverse the tree to find the appropriate concept (leaf).
3. If the number of observations within that leaf is greater than a threshold, subdivide it and create two children leaves.
4. Repeat.

The dimension along which to divide the subspace is that which most evenly divides the prior observations within that subspace. If the prior observations are all at one point within the subspace, subdivision would result in an empty subspace and is avoided. This encourages an economical conceptual representation of the state space. New concepts are not created for subspaces until they are visited repeatedly. It should be noted that although the decision tree uses a numerical vector, it does not impose any assumptions of metricity on the state space. That is, points that lie near each other, as defined by the Euclidean norm or any other p-norm, are not necessarily assumed to be more closely related conceptually. However, thanks to the numerical representation, if such an association does exist the decision tree can take advantage of it. Similarity between concepts (leaves on the tree) is measured by the number of nodes that must be climbed to find a common ancestor. An illustration of a simplie partitioning tree is shown in Figure 1.

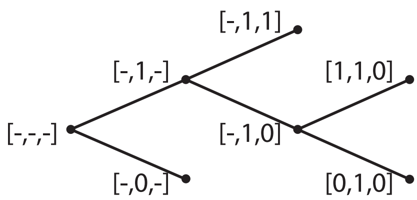

Figure 1: Schematic of a decision tree representing a concept hierarchy for a three-dimensional binary state space. The root node at the far left encompasses the entire state space. This is reflected in the fact that all three dimensions are unspecified. The first generation of nodes was created by dividing along the second dimension. In the first generation, the values of the second vector element indicate which branch corresponds to which half of the bisected dimension.

In addition to the fact that decision trees provide a natural structure for representing the repeated subdivision of the state space, they also are computationally efficient, allowing for searches in $O(\log n)$ time.

The decision tree approach reproduces an intuitive aspect of human psychological behavior: conceptual representation is most highly refined where it is most required. As an example, when first exposed to specific dogs we might conceptually group all dogs as *dog*. However, after further exposure, we are likely to generate more specific concepts such as *border collie*, *chow chow*, and *poodle*. Additional exposure may refine those concepts further to include *standard poodle*, *miniature poodle*, and *toy poodle*.

Robot implementation

The state space division algorithm described above was coded in Java and demonstrated with a Surveyor SRV-1 mobile robot (Surveyor Corporation, San Luis Obispo, California, USA). The SRV-1 is a tracked robot with a frontward-mounted color CCD and a Bluetooth radio. (Figure 2) It is relatively small, at 12 cm $\times$ 10 cm $\times$ 8 cm and weighs approximately 350 g. A video of the robot in action can be found at (Rohrer 2009b). The complete Java code used to control the robot, read in sensor data, and create the decision tree can be found at (Rohrer 2009a) in the package "becca_SRV".

The robot occupied a 102 cm $\times$ 72 cm room with black walls 40 cm high and a black floor. In the center of each

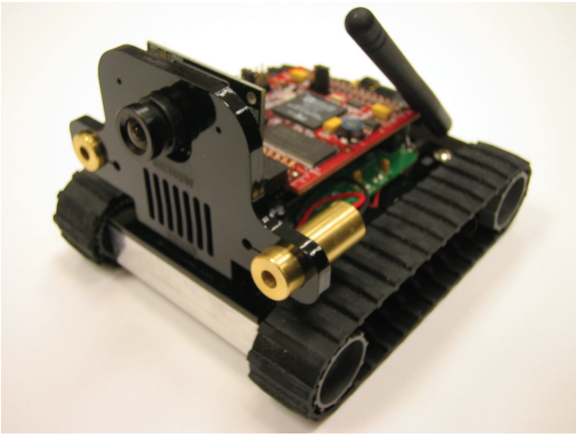

Figure 2: The Surveyor SRV-1 robot.

wall was a white stripe 13 cm wide extending the height of the wall. (Figure 3a) At each time step the robot returned an image from its camera to the controlling computer. (Figure 3b)

The color image from the robot was heavily pixelated to create 24 pixels, each with 5 grayscale values. Even at that coarse resolution, the state space contained 5^{24} ($> 10^{16}$) points, far more than could be explored in reasonable time. However, in practice the majority of those points would never be visited, and an even coarser grouping would be sufficient to allow the robot to achieve its goals.

Examination of the algorithm's performance during the robot's operation shows that the number of concepts created is indeed far smaller than the size of the state space, even for much reduced state spaces where full exploration is feasible. Specifically, binary dimensions that never change are not subdivided, resulting in an economy of representation.

Method II: Context-Based Similarity

A second concept creation method called Context-Based Similarity (CBS) groups states or symbols based on the states or symbols that temporally precede and follow them. (Hulet, Rohrer, and Warnick 2005) In contrast to state space partitioning, CBS creates concepts by agglomeration rather than subdivision. It creates concepts by groups of states, stimuli, or symbols that tend to occur in the same context. Here, "context" refers to the surrounding elements in the temporal sequence—those symbols that precede and follow the symbols of interest. The set of symbols that shares a given context constitutes a concept. This approach is well suited to high-level symbolic data in sequential streams and operates off-line. An example of CBS in a natural language processing application is shown in Figure 4. However, CBS uses no domain knowledge and would be equally applicable to other symbolic data, including language processing in Russian or Chinese, discretized and quantized audio or video data streams, or categorical data of any type.

Semantic clustering is the primary challenge in creating higher-level concepts from symbolic data. Once object clusters are formed (for instance, a Chevrolet Corvette, a Dodge

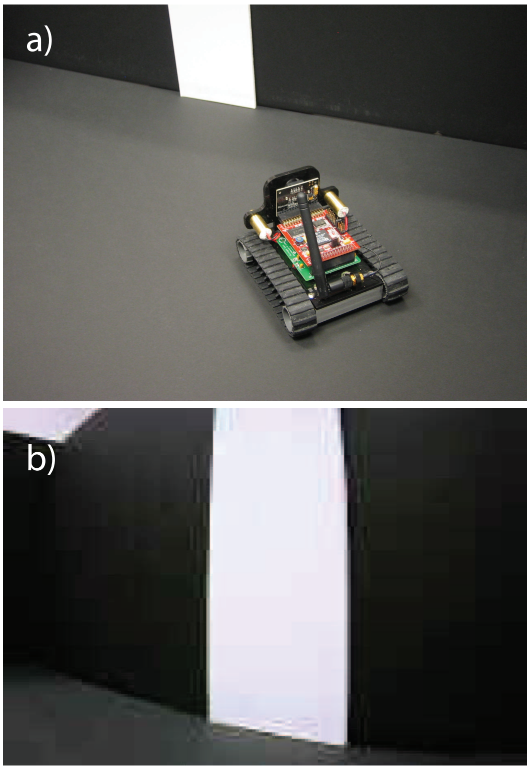

Figure 3: The room that served as the robot's environment. a) Viewed from above. b) Viewed from the robot's camera.

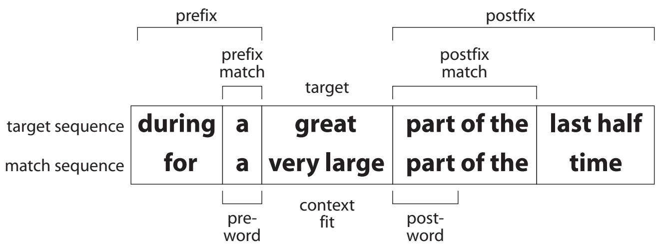

Figure 4: The similarity of two words or phrases can be judged by the amount of matching context. A target word ("great") is supplied. Word sequences containing the target are identified (e.g. "during a great part of the last half"). The target prefix ("during a") and postfix ("part of the last half") are identified as well. Sequences containing partial matches to the prefix and postfix are found (e.g. "for a very large part of the time"). The portion of the match sequence that falls between the partial prefix and postfix matches ("very large") is called the context fit and is determined to be somewhat related to the original target. When performed over a large sequence library, the strength of relation between a target and each context fit can be estimated based on the statistics of how many times each context fit is found and the length of the prefix and postfix matches in each case.

Viper, a Lamborghini Murcielago, and a Ferrari Enzo) then they can be conceived of as a collective, abstractly. If the higher-level concept is to be dealt with explicitly, a label (*sports cars*) is desirable, but not required. After the concept is created it can be expanded, modified, or re-combined with other concepts to create yet higher-level concepts. But at the root of this process is the ability to semantically cluster symbols (or objects or states) to create concepts.

At a crude level, CBS can be used to model human concept creation. If an enormous simplification is made, conscious perception can be approximated by a serial stream of discrete experiences. Those experiences can then be used in a manner analogous to that shown in Figure 4 to identify related experiences. For example, one day a child might hear the word "candy," see a gumdrop, then taste the sugar after putting it in her mouth. The next day she may have a similar experience, but with a piece of chocolate. In this case, the audible word "candy" would be the prefix and the sensation of tasting sugar the postfix. The gumdrop and the chocolate were both experienced between them and thus would come to be associated with each other. In this way, images of items of varying size and appearance can all become conceptually related. Perceptions across multiple sensor modalities can become associated in the same way.

CBS is different than the conceptual clustering methods CLUSTER/2, COBWEB, and UNIMEM in that it does not use any attribute data. CBS operates on symbol sequences only, and requires no descriptions, definitions, instances, or other elaborations of the meanings behind them. In contrast to some categorizing approaches, the concepts that CBS produces share important attributes with actual psychological conceptualizations. The membership of individual percepts within a concept is graded, rather than binary. (Labov 1973; Rosch 1975) Taking the concept of *bird* as an example, some animals are more birds than others. Despite the fact that there is a zoologically-defined class of animals that we refer to as birds, human subjects classify robins as more essentially bird than chickens. (Rosch 1973) Likely penguins and emus would be even less so. And the platypus, while not being a bird, may be classified as somewhat *bird* thanks to its duck-like feet and bill. A second aspect of CBS-generated categories that is consistent with psychology is that a single experience may have membership in many categories. A subtask within some standardized IQ tests illustrates this aspect of human concepts: name as many uses for a brick as possible within a fixed amount of time. Each separate use for a brick represents another concept of which a brick is a partial member. A few of the more mundane include *building material*, *paper weight*, and *projectile,* but there are far more creative uses as well. A percept may have membership in arbitrarily many CBS-generated concepts. And third, since concept membership is derived from experience, the concepts to which a percept belongs may change as additional experience is accumulated.

Natural language processing

CBS was applied to the problem of finding similar words and phrases in the English language, also called semantic clustering. No grammar, part-of-speech, or syntactic knowledge was used. First a sequence library was built by reading untagged text, using white space delimited words. To determine word similarity, the user first entered a target query. The sequence library was then searched for every sequence containing the target. For each target sequence the word immediately preceding the target (the pre-word) and the word immediately following the target (the post-word) were located. Next, the library was again searched, this time for every sequence containing the pre-word followed sometime thereafter by the post-word. These were match sequences. All phrases found occurring between the pre-word and the post-word in match sequences were context fits. This constituted a minimum requirement for contextual similarity. The two words flanking the target must also have been flanking any other context fit.

The similarity of context was judged by the number of additional words, or matches, common between the target and the context fit, beginning with the pre-word and post-word and counting outward (see Figure 4). Context fits were sorted by the number of matches; in the event of a tie, the context fits were sorted by the number of different contexts in which they appear. Two words found in a large number of shared contexts were usually more similar than two words which appeared in only one common context.

Java code that applies CBS to text processing, including the code for creating a sequence library from text files, can be found at (Rohrer 2009a) in the package "becca_text".

Results

The learner was tested after reading just over 25 million words from a diverse selection of texts. Figure 5 shows the first ten results returned on some sample queries, sorted by the above method. The results indicate that the learner is able to draw conclusions about strongly related words simply by observing how the language is used. The learner successfully generated groups of similar words using no predefined lookup tables, part-of-speech tags, or other previous knowledge of the language structure. The results were returned in an order intuitive to humans and suggest an accurate relative degree of similarity between words.

Antonyms, while clearly not synonyms, are strongly related word pairs, and appeared in similar contexts. Note that "small" ranks highly similar to "large". By allowing candidate context fits to be of any length, it was also possible to capture context fits requiring more than one word to express, such as "very large." These results support the idea that similar words appear in similar contexts. Consistently, the greater the contextual match up between the target and the context fit, the greater the similarity of the phrases.

Discussion

Both state space partitioning and CBS have demonstrated the creation new concepts, either from raw data or from lower-level concepts. The creation of concepts from data is not addressed often in cognitive architectures. Usually concepts are created explicitly by a human operator, most often during the programming of the system. In the SOAR architecture for example, a set of concepts are pre-defined

seven	large	sugar	feet	father	road
five	great	flour	face	mother	river
two	small	fruit	heart	wife	street
four	considerable	butter	head	son	table
three	certain	salt	side	head	fire
ten	very large	water	house	life	hill
twelve	good	mace	lips	voice	head
fifteen	very small	mear	work	face	house
twenty	vast	cream	hands	heart	room
fifty	larger	brandy	back	name	lake

Figure 5: The top ten semantic clustering results for some common terms in a 25M word corpus. In each case, CBS identified words and phrases that are semantically related to the target word. There are several details of the results that are noteworthy. For the target "seven," single digit numbers were identified as being the most similar, followed by larger numbers in roughly numerical order. Context fits for "sugar" included "mace" and "mear", two baking ingredients unknown to the authors at the time the test was run. The word "head" was matched to "father"–this is a figurative synonym, rather than a literal one.

by the programmer. Positive and negative training examples are hand-labeled by the programmer as well and used in a type of supervised learning. (Wray and Chong 2003) In these cases any concept learning that takes place involves mapping experiences onto pre-existing concepts. If an experience does not fit well with the set of concepts available to it, it is typically grouped with the best fit available, however poor. The creation of new concepts based on the system's experience, as demonstrated in this paper, is a deviation from this convention.

A note on the definition of "concept"

The definition used here of a concept as "a set of numerical or symbolic data that are closely related" is by no means a universally accepted one. While there are few definitions of "concept" that are specific enough to describe a unique algorithmic implementation in software, many definitions share two common elements. Primarily, a concept is a general class or idea inferred from specific instances. The definition used in this paper expresses this notion as well. But in addition to this, there is often a metaphysical element implied as well. This element is described variously as a quale (Lewis 1929), a Platonic form, an intentionality (Brentano 1874), or as the essence or the ideal of the thing being conceptualized. This aspect is not captured in the definition used in this work. This omission may reasonably be considered by some to weaken the claims to conceptualization and abstraction presented above. It is, at its heart, the same criticism made by Searle of his hypothetical Chinese Room–that although the room system may appear to understand Chinese to every external observation, it is in fact missing some critical mental element and does not. (Searle 1980)

While common, the idea that concepts comprise more than sets of specific experiences begs the questions "what else?" and "how does it come to be associated with experiences?" One possibility is that the essences of concepts exist prior to humans' experiencing instances of them. Perhaps they are part of neural circuitry at birth. This is consistent with the innateness hypothesis, which states that the human brain is hard wired at birth with specific aspects of language (Putnam 1967). Unfortunately, there is no description of the hypothesized "essence" of a concept specific enough to be scientifically verified or refuted, so the topic cannot be appropriately pursued in this forum. For the time being we will have to carry on with a definition of "concept" that is possibly partial, but at least concrete.

The definition of concept has direct relevance to the symbol grounding problem (Harnad 1990). As originally posed, the symbol grounding problem is the problem of assigning meaning to symbols. In other words, the problem of associating specific experiences with concepts. It has proved a difficult problem to address, but has attracted a good deal of attention, most often expressed as a supervised learning algorithm. However, the problem statement supposes that concepts precede the experiences. If it were the case that things worked the other way around, that is, that concepts were formed from experiences, then the symbol grounding problem would disappear. In fact, it has been argued that the view of the symbol grounding problem as the need to appropriately assign input data to a set of pre-determined symbols is backward, and that instead symbols should emerge from low-level sensor data. (Plunkett et al. 1992) This is precisely the approach taken by state space partitioning and CBS. State space partitioning creates concepts directly from low-level data, and CBS creates them lower-level concepts. In either case, the grounding is trivial.

Why concept creation?

The use of concepts in cognitive architectures uniquely aids them in performing cognitive tasks. Concepts allow economic reference to entire classes of objects, actions, or attributes. Reference to a general class, rather than to a specific object is the basis for abstraction in communication, language, and perhaps conscious thought. The sounds of the word "book" being spoken can become associated with the texture, heft, and appearance of a dictionary. The grouping of these three experiences, along with many others, into a single concept allows the communication of a rich set of multisensory experiences with a single spoken word. In addition, hierarchical conceptual representation allows specific experiences to be interpreted more generally. For instance, it allows the experience *this flying bullet that struck the wall behind my head just now* to become a generic *flying bullet*, which in turn becomes *a projectile*, *a physical assault*, and most generally, *an attack*. This in turn allows for the creation of highly efficient abstract communication, such as "when attacked, retaliate." In this way, high level abstractions also serve as building blocks for reasoning, analogy, rule creation, logic, and other cognitive activities.

The ability to acquire new concepts from experience has the potential to give machines far more learning capability than they currently possess. The fact that most cognitive architectures have a set of concepts fixed at compile time prohibits them from creating new representations of their experience. This narrows their task space. A chess-playing machine would not be expected to generalize well to an information retrieval task if it could only represent its world in terms of bishops, rooks, and board positions. Creating concepts based on a system's experience allows application to a much broader task space and helps to ensure that its concepts are of appropriate detail. It also does not rely on the programmers' foresight to create a complete list of relevant concepts. Concept creation helps to bridge the divide between data-driven, reactive robots and systems that perform sophisticated reasoning tasks.

Another way of expressing the limitation of fixed conceptual representation is that it introduces a large set modeling assumptions. Depending upon the number and specificity of the concepts provided, those modeling assumptions may introduce minor or major limitations on the system's performance. But providing the capability to create new concepts greatly decreases the constraints on what the system can ultimately learn to do. Cognitive architectures striving for human-level generality will not be likely to achieve it in a system that cannot create new conceptual representations of its experiences as it learns.

Acknowledgements

This work was supported by the Laboratory Directed Research and Development program at Sandia National Laboratories. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.

References

Bentley, J. L. 1975. Multidimensional binary search trees used for associative searching. *Comm. ACM* 18:509–517.

Breiman, L.; Friedman, J. H.; Olshen, R. A.; and Stone, C. J. 1984. *Classification and Regression Trees*. Belmont, California: Wadsworth International Group.

Brentano, F. 1874. *Psychology from an Empirical Standpoint (Psychologie vom empirischen Standpunkt)*. Liepzig: Verlag von Duncker and Humblot. Available from Google Books: http://books.google.com/books?id= AyUAAAAAQAAJ&oe=UTF-8.

Carbonell, J.; Klein, S.; Miller, D.; Steinbaum, M.; Grassiany, T.; and Frey, J. 2006. Context-based machine translation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, 19–28.

Fisher, D. H. 1987. Knowledge acquisition via incremental conceptual clustering. *Machine Learning* 2:139–172.

Harnad, S. 1990. The symbol grounding problem. *Physica D* 42:335–346.

Hulet, S.; Rohrer, B.; and Warnick, S. 2005. A study in pattern assimilation for adaptation and control. In *8th Joint Conference on Information Systems*.

Labov, W. 1973. *New ways of analyzing variations in English*. Washington D. C.: Georgetown Univeristy Press. chapter The boundaries of words and their meanings, 340–373.

Landauer, T. K., and Dumais, S. T. 1997. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review* 104:211–240.

Lebowitz, M. 1987. Experiments with incremental concept formation: Unimem. *Machine Learning* 2:103–138.

Lewis, C. I. 1929. *Mind and the World Order*. New York: Scribner's Sons.

Lund, K.; Burgess, C.; and Atchley, R. A. 1995. Semantic and associative priming in a high-dimensional semantic space. In *Proceedings of the Seventeenth Annual Meeting of the Cognitive Science Science Society*, 660–665. Hillsdale, NJ: Erlbaum.

McCallum, R. A. 1996. Hidden state and reinforcement learning with instance-based state identification. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics* 26(3):464–473.

Michalski, R. S., and Stepp, R. 1983. Automated construction of classifications: Conceptual clustering versus numerical taxonomy. *IEEE Trans on Pattern Analysis and Machine Intelligence* 5(4):396–410.

Miller, G., and Charles, W. 1991. Contextual correlates of semantic similarity. *Language and Cognitive Processes* 6(1):1–28.

Plunkett, K.; Sinha, C.; ller, M. F. M.; and Strandsby, O. 1992. Symbol grounding or the emergence of symbols? vocabulary growth in children and a connectionist net. *Connection Science* 4(3 and 4):293–312.

Putnam, H. 1967. The 'innateness hypothesis' and explanatory models in linguistics. *Synthese* 17:12–22.

Quinlan, J. R. 1987. Simplifying decision trees. *International Journal of Man-Machine Studies* 27:221–234.

Rohrer, B. 2009a. Becca sourceforge repository. http://becca.sourceforge.net.

Rohrer, B. 2009b. Robot learning with a biologically-inspired brain (becca). http://www.youtube.com/watch?v=mJ8LMMHsUf8.

Rosch, E. H. 1973. Natural categories. *Cognitive Psychology* 4:328–350.

Rosch, E. H. 1975. Cognitive representation of semantic categories. *Journal of Experimental Psychology* 104(3):192–233.

Rubenstein, H., and Goodenough, J. B. 1965. Contextual correlates of synonymy. *Computational Linguistics* 8(10):627–633.

Searle, J. 1980. Minds, brains, and programs. *Behavioral and Brain Sciences* 3(3):417–457.

Wang, X.; McCallum, A.; and Wei, X. 2007. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM)*.

Wray, R. E., and Chong, R. S. 2003. Explorations of quantitative category learning iwht symbolic concept acquisition. In *Proc 5th Intl Conf on Cognitive Modeling (ICCM)*.